Computing with confidence: A Bayesian approach

D. Partridge, J.E. Fieldsend, W.J. Krzanowski, T.C. Bailey, R.M.Everson and V. Schetinin*
School of Engineering, Computer Science and Mathematics
University of Exeter, Exeter EX4 4QF

*Department of Computing, University of Luton, Park Square, Luton, Beds. LU1 3JU, UK.

{D.Partridge[a],J.E.Fieldsend,W.J.Krzanowski,T.C.Bailey,R.M.Everson,V.Schetinin}@Exeter.ac.uk

Abstract
Bayes' rule is introduced as a coherent strategy for multiple recomputations of classifier system output, and thus as a basis for assessing the uncertainty associated with a particular system results --- i.e. a basis for confidence in the accuracy of each computed result. We use a Markov-Chain Monte Carlo method for efficient selection of recomputations to approximate the computationally intractable elements of the Bayesian approach. The estimate of the confidence to be placed in any classification result provides a sound basis for rejection of some classification results. We present uncertainty envelopes as one way to derive these confidence estimates from the population of recomputed results. We show that a coarse *SURE* or *UNSURE* confidence rating based on a threshold of agreed classifications works well, not only pinpointing those results that are reliable but also in indicating input data problems, such as corrupted or incomplete data, or application of an inadequate classifier model.

Introduction

Every significant computational system is likely to contain errors. In the current context, i.e. a classifier system that is generated by tuning the parameters of a classifier model (such as a neural network) using a set of data samples (the training set), error-free computation, in the sense of correct classification for every conceivable input, may not be possible, even in principle. The training data may be corrupted, or incomplete, or it may derive from a problem in which the target classes overlap to some degree.

The traditional approach to quantifying such computational uncertainty is to test the optimized classifier system on a large test set of data, and to generate an average uncertainty or confidence in the system performance. Thus if the system fails on, say, 3% of test samples we might begin to conclude that our classifier is approximately 97% reliable. The ways in which such results are interpreted are not important. What is important is that the traditional approach to quantifying software system reliability or uncertainty is by means of *global* system estimates; we obtain a measure that relates to the average system performance. One view of what we are proposing is a switch to *localized* estimates of system reliability which may involve an decrease in overall system reliability, but is a more useful strategy simply because meaningful reliability, through confidence estimates, accompanies every computed result, and is specific to the result, not merely a global average.

The strategy presented below arises as a by-product of the Markov Chain Monte Carlo (MCMC) process that has been developed in recent years for implementation of Bayesian Models in general (Green, 1995, MacKay, 1998) and Bayesian classification in particular (Denison *et al,* 2002). By adopting a Bayesian approach we average across thousands of individual classifier results selected probabilistically by the MCMC process according to the Bayesian *a posteriori* individual model probabilities. But the multiple computations are not used solely to improve overall classifier performance per se (i.e. percent correct), they are also used to generate an accurate estimate of the confidence that can be placed on the correctness of each classification result.

If a majority of the classifiers agree on the classification result, then that result will receive a high confidence. The proportion in agreement of the population sampled is used to estimate confidence, and the

[a] corresponding author

fact that we sample thousands of classifiers, each selected systematically, means that we have a sound basis for confidence estimation.

Bayesian averaging

We consider a classifier system where there are Q target classes denoted by $Y = (c_1, \ldots, c_Q)$, a set of predictor variables $x_n = (X_1, \ldots, X_p)$, a training data set *D*, and a classifier model $C(x_n, \theta)$ that depends on the predictors as well as on a set of parameters θ. For real data sets any classifier model applied is necessarily an approximation to the truth, and it is typically the case that within the chosen classifier-model type (e.g. k nearest neighbours or multilayer perceptron) further choices must be made of values for the parameters θ of the model. The classical single-classifier approach replaces θ by an estimate (e.g. maximum likelihood or least squares) derived from the data set *D*.

We now briefly outline the Bayesian approach to classification, which permits incorporation of any *a priori* knowledge that is available. For a Bayesian approach we need to specify a joint prior distribution $\pi(\theta)$ for the classifier parameters, to derive the likelihood $L(D \mid \theta)$ of the training data using an appropriate probability model (just as in the classical approach), and hence obtain the posterior distribution of the parameters,

$$\pi(\theta \mid D) \propto \pi(\theta) L(D \mid \theta)$$

where the constant of proportionality *b* is chosen to make this posterior distribution integrate to 1 over the range of θ. The Bayesian classifier is then the expected value of $C(x_n, \theta)$ over this posterior distribution (obtained by integrating $C(x_n, \theta)\, \pi(\theta \mid D)$ over the range of θ). Evaluating the two integrals can be computationally very difficult, particularly when the dimensionality of θ is large, and this hampered the development of Bayesian approaches to classification for many years. However, the Bayesian classifier can be well approximated by the mean of $C(x_n, \theta)$ over a large sample of independent observations from $\pi(\theta \mid D)$, and MCMC enables samples to be drawn from a distribution which is known up to a constant of proportionality (thus eliminating the need to evaluate *b* by integration). Each step of the process is a transition of a Markov Chain whose probabilities are arranged so that $\pi(\theta \mid D)$ is the limiting (stationary) distribution. The process thus requires us to run the chain for a preliminary (burn-in) period to ensure stationarity has been reached, and then to sample (say) every 7th value to ensure independence of observations. Each observation then yields a single classifier whose outcomes on the test set can be stored. The average of the complete set of classifiers gives the Bayes classifier, and the individual outcomes are used in to derive the confidence measures as indicated below.

Within the framework of a well-constructed MCMC approach to classifier sampling over large numbers of classifiers, we anticipate accurate estimation of the posterior probabilities necessary for Bayesian averaging. Consequently, our method can be seen as an advance on previous (and self-styled "naïve") approaches to Bayesian averaging (e.g. Kuncheva and Whitaker, 2001; Xu, Krzyzak and Suen, 1992).

In this paper we focus on a version of the k nearest neighbour model, a probabilistic form (pknn). The treatment of the pknn model is based on Denison et al. (2002) where they define the likelihood of the data given the parameters θ as β and *k*:

$$L(D \mid \beta, k) = \prod_{i=1}^{n} \frac{\exp((\beta/k) \sum_{j \sim i}^{k} \delta_{y_i y_j})}{\sum_{q=1}^{Q} \exp((\beta/k) \sum_{j \sim i}^{k} \delta_{q y_j})}.$$

Here *k* is the number of neighbours that influence the prediction, y_i is the class membership of the *i*th object, β is a parameter that controls the strength of association between the neighbouring y_is, δ_{ab} is a

Kronecker delta (i.e. equals one if a=b, zero otherwise) and $\sum_{j\sim i}^{k}$ denotes the summation over the k nearest neighbours of x_i in the set of the observed predictor locations not including the ith one. The term $\frac{1}{k}\sum_{j\sim i}^{k}\delta_{qy_j}$ records the proportion of points of class q in the k nearest neighbours of x_i.

This leads (from Holmes and Adams, 2002) to the predictive distribution for the class membership of x_{n+1} (a test point) following from conditional probability and being given by $pr(y \mid x,\beta,k,D) = \dfrac{\exp((\beta/k)\sum_{j\sim n+1}^{k}\delta_{y_{n+1}y_j})}{\sum_{q=1}^{Q}\exp((\beta/k)\sum_{j\sim n+1}^{k}\delta_{qy_j})}$

Treating β and k as known and fixed, or varied in some heuristic way, fails to account coherently for the uncertainty of these model parameters. In a Bayesian setting this uncertainty can be accommodated provided a joint prior for β and k is specified. In the absence of specific knowledge about the likely values of β and k, other than the fact that both should be positive, the following default prior densities are suggested:
π (k)=$U\{1,...,k_{max}\}$, k_{max}=min(n,200) independently of π (β)=$U(0,\infty)$.

A standard MCMC strategy is used to select classifiers determined by particular parameter values as an approximation to an integration over all possible values. After an initial burn-in period (typically 10,000 steps in the chain), any proposed move to a new classifier model (i.e. new values of β and k) from the current model (β,k) to the proposed model (β',k') is accepted if u, a draw from a $U[0,1]$ distribution, is less than $\min\left\{1,\dfrac{L(D\mid\beta',k')}{L(D\mid\beta,k)}\right\}$ otherwise the current values of β and k are retained.

The variety of the pknn model used expands the β parameter into a scaling matrix of parameters in order to accommodate variable feature-association strengths.

Classifier system performance

As introduced above, we extend the idea of classifier system performance enhancement beyond simply percent correct (or some variant dependent upon problem specific variation in the importance of the alternative classifications). We localize the idea of system reliability by generating a confidence associated with every system classification result. A Bayesian average result will, in general, be no worse in terms of percent correct classification performance than the 'best' model within the set sampled. In this study, we will use the maximum a posteriori (MAP) classifier from each set of samples to illustrate this point.

For our purposes a classification result is an ordered pair *(class,confidence)* within which the first element is the classification generated and the second is an estimate of the confidence to be placed in this particular result. As such, this study can be viewed as a contribution to the wealth of work on classifier acceptance-reject rates (e.g., Giacinto, Roli and Bruzzone, 2000) although this is not the emphasis of this paper. The element *confidence* might be a real value in the interval (0,1) where 0 signifies total lack of certainty in the classification result, and 1 signifies total certainty. It may also be useful to consider *confidence* as a categorical variable with, say, two values *SURE* and *UNSURE* in which case a reject threshold must be determined, and the *UNSURE* results are those that are rejected.

In this paper we will set an arbitrary threshold to determine the *SURE* and *UNSURE* categories. Exactly what threshold is set and precisely how the raw confidence values are computed will be pursued only as far as is necessary to support the contention that both decisions are reasonable options from a large number of equally reasonable alternatives. The focus of this study is presenting an existence `proof' that meaningful

localized reliability estimates can be generated. Thus the essential components of performance evaluation are: correct *SURE, UNSURE*, and incorrect but *SURE* classifications, which will total 100%.

The *SURE* and *UNSURE* confidence estimates

We set the threshold for the *SURE*/*UNSURE* boundary at 99% --- if the raw confidence value associated with a classification result is less than or equal to 0.99 then that classification is *UNSURE*. The raw confidence values derive from an *uncertainty envelope* around the problem decision boundary whose perimeter is established by the proportion of correct (i.e., correct class assigned a probability >0.5) to incorrect classifications on each training data point given by the sample of classifiers selected by the MCMC approach. In the current case, the boundary of the uncertainty envelope is given by that subset of the training data for which exactly 0.01% of the classifiers sampled yielded an incorrect classification result.

The 99% threshold is entirely arbitrary but the uncertainty envelope approach to confidence estimation has been justified elsewhere (Fieldsend et al., 2003) as an improvement on other approaches, such as error bars (Bishop, 1995).

Results

Our MCMC based Bayesian approach was applied to seven data sets: one a synthetic set devised by Ripley (Riply, 1994) and augmented with a further Gaussian function (making five gaussians --- 3 contributing to one class, and 2 to the other, full details in Fieldsend et al., 2003) , and called the Synthetic data. The other six sets are from the UCI Machine Learning repository; they are: Wisconsin, Ionosphere, Votes, Sonar, Vehicle and Image. The data-set details and the classification results are given in Table 1. Bold indicates better values than MAP (or vice versa).

Table 1: Classifier system results on various test problems

					MAP	Bayesian average			
Data	Q	p	D	Test Set size	Correct (%) **A**	Correct (%) **B**	*SURE(%)* Correct **C**	*UNSURE* (%) **D**	*SURE(%)* Incorrect **E**
Wisconsin	2	9	455	228	*98.3*	**99.1**	88.6	11.4	0.0
Ionosphere	2	33	200	151	*93.4*	**94.0**	58.9	41.1	0.0
Votes	2	16	391	44	*93.2*	**95.5**	81.8	15.9	2.3
Synthetic	2	2	250	1000	**89.5**	*88.8*	83.3	10.9	5.8
Sonar	2	60	138	70	*81.4*	**88.6**	20.0	80.0	0.0
Vehicle	4	19	564	282	*64.2*	**67.7**	47.5	42.6	9.9
Image	7	19	210	2100	14.3	14.3	0.0	100.0	0.0

Discussion and Conclusions

First we note that in every case illustrated in Table1, the global estimate of system reliability (col. **B**) appears to be higher (sometimes very much higher) than system reliability estimated using our localized approach (col. **C**). However, with our confidence scheme, system reliability emerges in a new guise. Consider the Wisconsin data, for example, column **B** suggests that the Bayesian average system can be expected to compute the correct class on about 99 attempts in every 100. So we can expect a wrong classification result now and again, but we have know idea which particular computations will be incorrect. We just know that on average 1 in 100 computations is likely to deliver an incorrect classification. Contrast this type of knowledge of system uncertainty with that available through our confidence estimates and given in columns **C**, **D** and **E**. In this latter case, we must expect only about 89% correct computations, but the *SURE* confidence label associated with each one tells us exactly which computations are correct and

which are not. In a very real sense, we might claim that for the Wisconsin data our confidence system is, in fact, 100% reliable, because we expect the *SURE*/*UNSURE* label with every computed result to tell us which are correct and which are not. This somewhat extreme interpretation of our results does raise a note of caution: if we attach complete certainty to all our confidence labels then serious problems may arise when that certainty is misplaced, as it would be for any data set that does not exhibit a zero in column **E**. Even for those data sets that do exhibit zero in column E, such as the Wisconsin data, it would be foolish in the extreme to assume that the system, which performed well under the test whose results are presented, will never produce a *SURE* but incorrect result.

The nature of software testing demands that a certain degree of circumspection be attached to interpretation of the system uncertainty revealed. The global average estimate, the traditional approach, inherently includes a caution with every application of the tested system, but our localized approach could lead the naïve user to assume total reliability, as suggested with the Wisconsin data result. We do not advocate this, but neither do we think that our scheme leaves the user with much the same problem of system output interpretation. Our scheme does not banish uncertainty (this will never be done, even with proven algorithms). What it does is to refine system uncertainty estimation with respect to specific system outputs. How exactly the user should use the richer uncertainty estimation offered must be dependent upon particular systems and their particular applications.

We also note that the performance of the `best' (i.e. MAP) classifiers is, in general, worse than the Bayesian average. There is, however, an exception: the Synthetic data. Such exceptions are to be anticipated as artefacts of the specific test applied. The bayes average performances, being averages of 10,000 classifiers, will, in general, exhibit greater stability with respect to test-set variation. In addition, it should be noted that all 10,000 classifiers must be generated as a prerequisite to selecting the best.

Any claims rest on the assumption that the observed values are significant to the order of a few percent. This raises the problems of accuracy assessment, an issue that is usually approached through repetition of experiments. Repartitioning of the data is one route to such repetition. Although we have used only the `standard' data partitions given in the UCI repository, it should be noted that each one of our classifications results is an 'average' of 10,000 individual results, and for the Synthetic, for example, the standard deviation of the 10,000 confusion matrices generated from the individual classifiers sampled is of the order of 0.7%. This is further evidence that the averages presented will support significance claims well within the few percent that are required for current arguments.

TheTable contains point estimates of percent correct. In order to draw precise firm conclusions about the relative merits of the strategy proposed (rather than just demonstrate general feasibility), it is desirable to obtain estimates of the variability of all these values. Each estimate with an associated confidence derives from 10,000 classifications of each test point only in the sense that 99% or more of the test points agree on a classification (i.e. *SURE*), or do not (i.e. *UNSURE*). Ultimately, the only route to variability estimation for the confidence values assigned is to run the Markov Chains longer to obtain further samples of, say, 10,000 classifications, and so obtain a set of estimated confidences to provide a basis for calculating standard deviation, or some such measure.

One possible result of this large escalation of model complexity, and hence model-space to be sampled, is that for such highly complex models the 10,000 samples generated will contain more than 100 (i.e. 1%) of highly suboptimal classifiers which will lead to an increase in *UNSURE* classifications (such as we see in some data-set results). If this is indeed the reason behind the high percentage of *UNSURE* classifications detailed, then lowering the threshold for *SURE/UNSURE* from 99% will result in a transfer of classifications from the *UNSURE* to the *SURE* correct category. There is some evidence to support this contention: when the threshold for *SURE*/*UNSURE* is lowered for the Sonar data, we find a trend of transfer of results from *UNSURE* to (primarily) *SURE* correct. Thus when the threshold is set at 80% (rather than the standard 99%), nearly half of the *UNSURE* results, i.e. 37%, become reclassified as 36% *SURE* correct (making 56% *SURE* correct in total --- see row 5 in the Table) and 1% *SURE* incorrect.

The fact that the Image data does not yield any *SURE* classifications may be due to the (relatively) large number of target categories (Q=7) it involves. An argument that implies the same remedy as presented

above is that the Image data offers an increased number of ways to wrongly classify a data point. Consequently, it is relatively more unlikely (than when fewer target categories) that wrong classifications will be limited to just 1% of classifiers sampled, and on the evidence in the Table this situation never occurs. So, in this case, lowering the *SURE/UNSURE* threshold should result in a transfer of test cases from *UNSURE* to either of the *SURE* outcomes. In this case, initial empirical probes provide no evidence to support this analysis: even when the *SURE*/*UNSURE* threshold is reduced to 50% for the Image data, 100% *UNSURE* results are still the outcome. The necessarily tentative conclusion for the Image data to explain its poor showing in the Table must be that either the pknn classifier is a poor model for this data, or that this data is insufficient or incomplete for the desired classification task, or both.

The overall conclusion to this study is that it is possible, and feasible given current levels of available computer power, to use this massive-recomputation strategy to effectively localize system uncertainty. But this enrichment of system uncertainty knowledge, although promising, demands a more sophisticated interpretation of the in-use system performance.

Acknowledgements

The authors gratefully acknowledge grant GR/R24357/01 from the EPSRC that funded this research under the Critical Systems Programme.

References

Bishop, C.M. 1995 *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

Denison, D.G.T., Holmes, C.C., Mallick, B.K. and Smith, A.F.M. 2002. *Bayesian Methods for Nonlinear Classification and Regression.* John Wiley & Sons, Ltd., Chichester.

Fieldsend, J.E., Bailey, T.C., Everson, R.M., Krzanowski, W.J., Partridge, D. and Schetinin, V. 2003 "Bayesian Inductively Learned Modules for Safety Critical Systems," Proceedings of the 35th Syposium on the Interface: Computing Science and Statistics. March 12-15, Salt Lake City.

Giacinto, G., Roli, F. and Bruzzone, L. 2000 "Combination of neural and statistical algorithms for supervised classification of remote-sensing images," *Pattern Recognition Letters* **21**, 385-397.

Green, P.J. 1995 "Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination," *Biometrika,* **82**, 711-732.

Holmes, and Adams 2002 ????

Kuncheva, L.I. and Whitaker, C.J. 2001 "Feature subsets for classifier combination: an enumerative experiment," Springer-Verlag Lecture Notes in Computer Science 2096, J.Kittler and F.Roli (eds), 228-237.

MacKay, D.J.C. 1998 "Introduction to Monte Carlo methods," In: *Learning in Graphical Models,* M.I. Jordan (ed), 175-204. Kluwer Academic Press.

Partridge, D. and krzanowski, W.J. 1997 "Software diversity: practical statistics for its measurement and exploitation," *Information and Software Technology* **39**, 707-717.

Ripley, B.D. 1994 "Neural networks and related methods for classification," *Journal of the Royal Statistical Society B*, **56**(3), 409-456.

Xu, L., Krzyzak, A. and Suen, C.Y. 1992 "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. on Systems, Man, and Cybernetics* **22**, 418-435.